
Rapport du projet « Réseau social pour étudiants », le 22 mai 2025

Groupe1
DIAS Flavien
SEGOT-LABEROU Maxime
BIRET-TOSCANO Esteban



Projet GitHub : <https://github.com/EstebanBiret/SoStudy>

Sommaire :

I. Étapes de la présentation	2
II. Schéma de la BD & DC	3
III. Organisation du projet	4
IV. Insights du projet	7
V. Conclusion	8

I. Étapes de la présentation

Mise en contexte de l'application, rappel des attentes et des fonctionnalités de l'application.

Démonstration de l'application en temps réel :

- 1^{ère} étape : création d'un compte puis déconnexion et reconnexion.
- 2^{ème} étape : suivre un utilisateur.
- 3^{ème} étape : reposter et mettre une réaction à un post de cet utilisateur.
- 4^{ème} étape : commenter un post.
- 5^{ème} étape : voir la page d'accueil.
- 6^{ème} étape : rejoindre et quitter une communauté.
- 7^{ème} étape : voir les posts d'une communauté.
- 8^{ème} étape : créer et modifier une communauté.
- 9^{ème} étape : rédiger un post pour cette nouvelle communauté.
- 10^{ème} étape : voir son profil, ses abonnements et modifier son profil.
- 11^{ème} étape : voir les événements, participer à un événement/ne plus y participer, organiser un événement et le modifier.
- 12^{ème} étape : voir la messagerie, créer une discussion, envoyer un message à un autre utilisateur.
- 13^{ème} étape : se connecter en navigation privée avec cet autre utilisateur, et voir les messages en temps réel de l'utilisateur précédent.
- 14^{ème} étape : créer un groupe, le modifier et quitter un groupe.
- 15^{ème} étape : supprimer le compte.

II. Schéma de la BD & DC

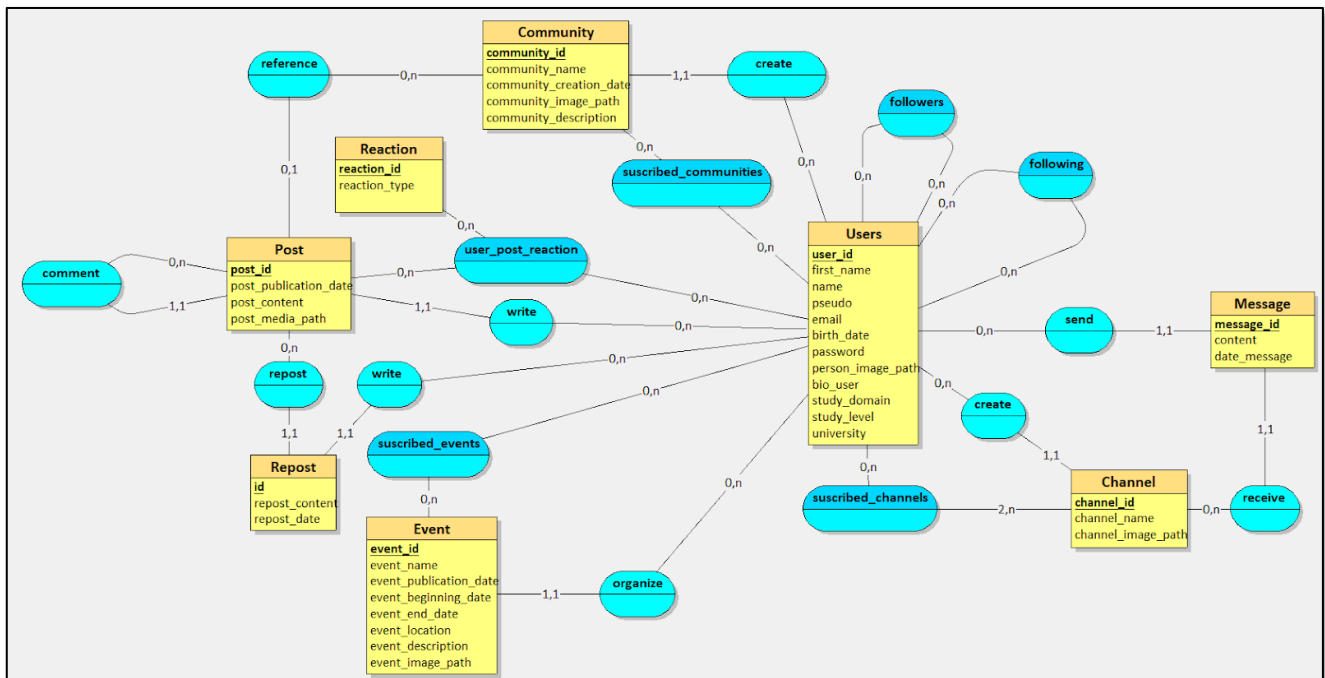


Figure 1 : schéma de la base de données

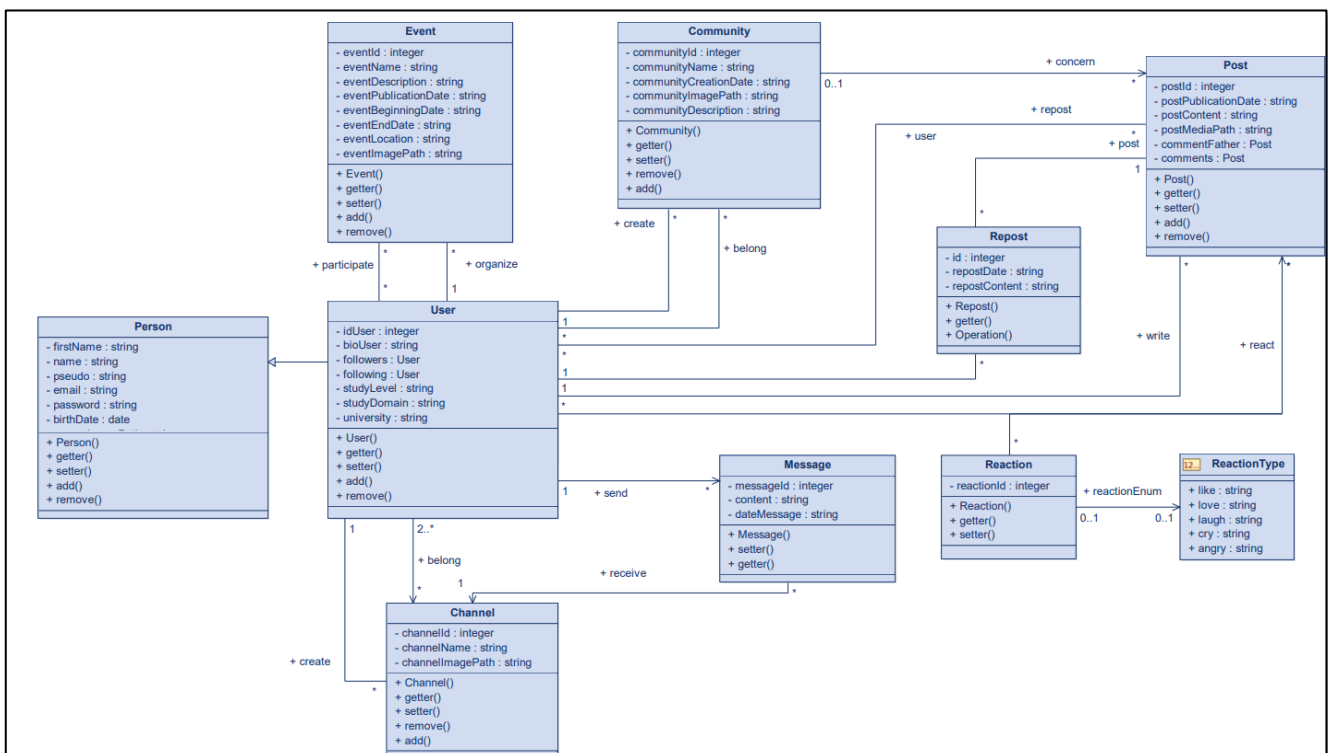


Figure 2 : diagramme de classes

III. Organisation du projet

Au début du projet, nous avons discuté ensemble afin d'évaluer la difficulté générale du projet. Nous avons découpé les fonctionnalités du réseau social d'étudiants en plusieurs **User Stories**, et pour chaque User Story nous avons mis en place des **scénarios de test**.



Figure 3: une US type



Figure 4 : un scénario de test type

Ensuite, nous avons estimé en heures la durée de chaque US, et commencé à réfléchir au schéma de la base de données et au diagramme de classes. Pour finir, nous avons ajouté des **dettes techniques** au Kanban, qui correspondaient aux tâches qui étaient nécessaires avant de commencer réellement à développer les fonctionnalités de l'application (création des entités, de la barre de navigation, et du système d'inscription/connexion).

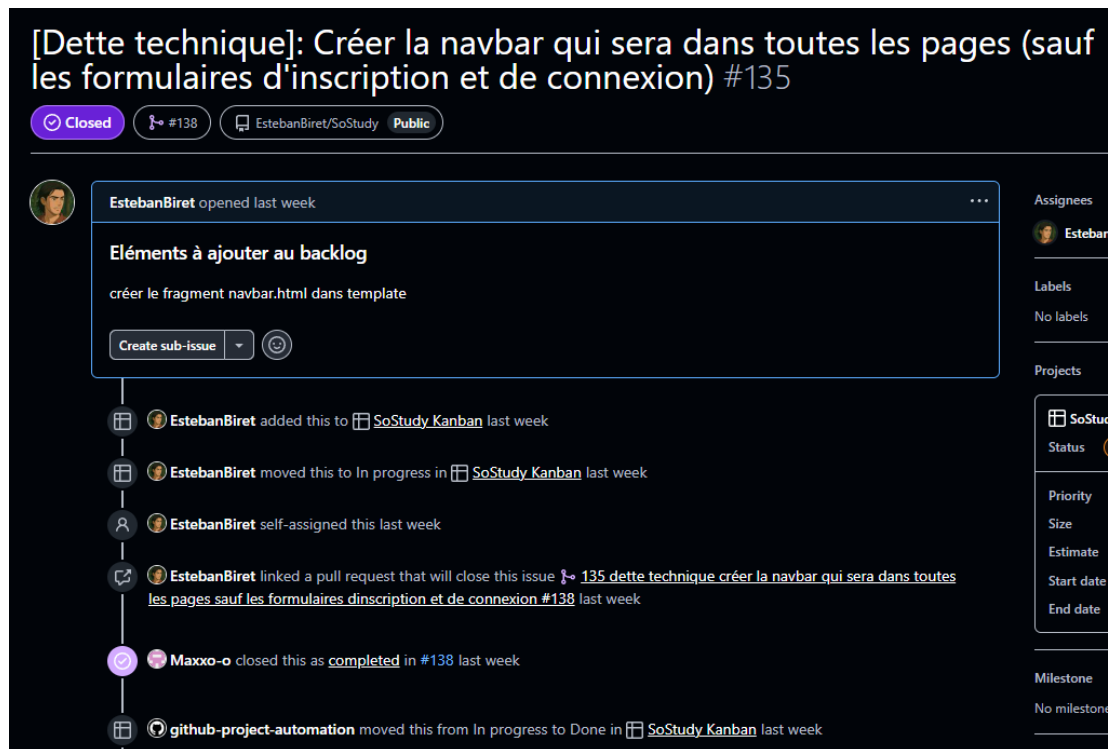


Figure 5 : une dette technique type

Maxime s'est occupé de réaliser quelques maquettes et la charte graphique du site (Cf. le dossier **/docs** dans notre repository GitHub). Nous nous sommes ensuite répartis les fonctionnalités par **grande thématique**, afin de travailler sans interférences et de garantir un développement efficace. Maxime s'est mis sur le système de messagerie et de gestion de groupes (création de discussion à 2 et groupe à plusieurs, envoi de messages en temps réel via **WebSocket**, modification du groupe pour le créateur, et possibilité de quitter un groupe). Flavien sur la partie des profils (inscription, connexion, déconnexion, page du profil, abonnement, désabonnement et recherche de profils). Esteban s'est penché sur le feed principal avec les posts (publication et suppression), les commentaires, qu'ils soient directs ou des réponses, reposts et annulations de reposts, ajout, modification et suppression de réactions, et les événements et communautés (les créer, les modifier, les supprimer, les quitter et les rejoindre.) Il est également possible de visualiser la liste des posts d'une communauté en cliquant sur les détails.

Durant tout le développement, nous avons veillé à faire de la javadoc et à respecter la qualité de code via les indicateurs de **Sonar**, seule la couverture de test est insuffisante (nous avons suivi les indications de M. Navarre, qui consistaient à tester uniquement les entités, ce qui explique notre taux de couverture de code de 22%).

Avant chaque développement d'une nouvelle fonctionnalité, nous procédions ainsi :

- Annonce de la fonctionnalité au reste de l'équipe.
- Sur le Kanban, attribution de l'issue et déplacement de cette issue vers la colonne 'In progress'.
- Création d'une branche à partir de cette issue
- Développement de la fonctionnalité
- Et enfin, création d'un pull request, et revue de code avec le reste de l'équipe avant de merge la pull request dans le main.

Chaque matin, nous faisons une réunion (**daily meeting**), afin de rapidement discuter de nos avancées, nos difficultés et nos prévisions pour les fonctionnalités à développer. Également, chaque début de semaine, les US étaient triées par ordre de priorité dans le backlog, et nous déplaçons les US à développer dans la semaine dans la colonne 'Ready'.

A la fin du projet, nous avons développé toutes les fonctionnalités que nous avions initialement prévu, sauf le système d'administration :

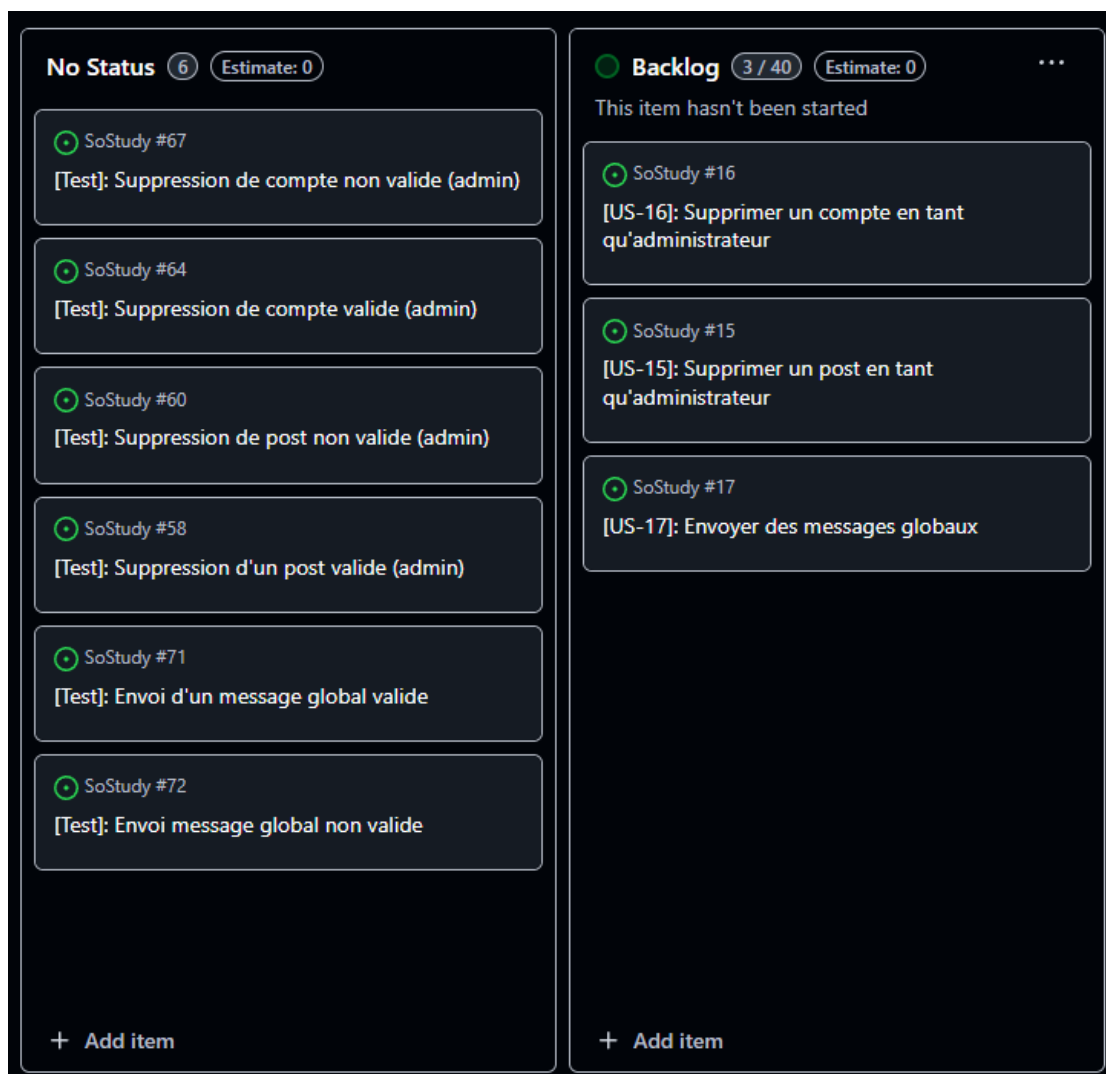


Figure 6 : backlog en fin de projet

Pour ce qui est des difficultés, elles se trouvaient principalement au début du projet, car nous n'avions pas encore harmonisé nos conventions développement, que ce soit concernant le nommage des attributs, des fonctions, la rédaction de la javadoc, ou l'approche des tests unitaires. Nous avons estimé que le délai de deux semaines était un peu court pour avoir une application propre, intuitive, bien testée et conforme aux attentes.

IV. Insights du projet

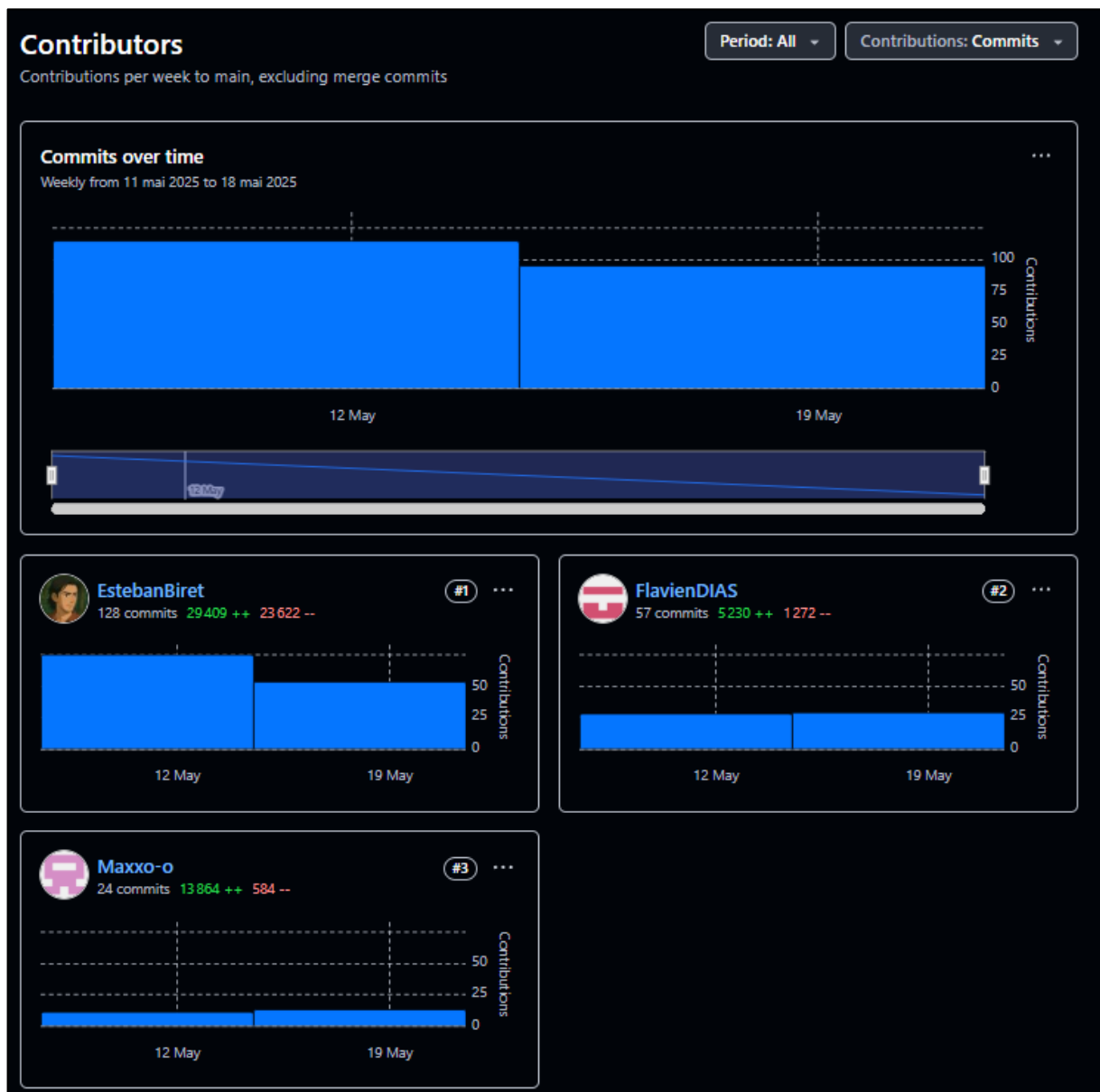


Figure 7 : insights du projet

On peut voir qu'Esteban a effectué davantage de commits, car il a travaillé sur un large panel de fonctionnalités différentes, et donc des commits réguliers étaient nécessaires pour bien séparer les fonctionnalités et avoir le code à jour. Maxime en a réalisé moins car il s'est concentré sur moins de fonctionnalités, mais qui étaient plus compliquées à développer.

V. Conclusion

Ce projet de réseau social pour étudiants nous a permis de mobiliser des compétences variées en conception, développement, gestion de projet et collaboration d'équipe. Malgré un délai relativement court, nous avons su nous organiser efficacement pour atteindre nos objectifs principaux. Le travail en équipe, bien que parfois confronté à des divergences de pratiques, s'est rapidement structuré grâce à une bonne communication et à une répartition claire des tâches. Nous retenons de cette expérience une meilleure maîtrise des outils de développement collaboratif, une compréhension plus fine des architectures web complexes, et surtout, l'importance d'une rigueur constante dans l'application des bonnes pratiques de codage. Si certaines fonctionnalités, comme le système d'administration, n'ont pas pu être intégrées à temps, le socle de l'application est solide, fonctionnel et répond aux besoins initiaux.